Feedback Tuning with Step Response

Step response is often used as a method of evaluating a feedback filter. Many controls textbooks contain information on interpreting step responses for establishing proper feedback, particularly for second-order systems (current-controlled motors driving inertial loads are second-order systems). In a step response, a sudden change is made to the command position, and the feedback filter attempts to bring the system to this new position. In observing how the system gets to the new position, we can deduce a great deal about the properties of the system. It does not matter that you will not ever create such a large instantaneous step in position in the actual operation of your system. The purpose of this "jolt to the system" is to bring out system characteristics that might otherwise not be obvious. This detailed information on the PID filter is not essential to performing the tuning, but is included here for references.

PMAC has three feedback parameters to be adjusted in this process:

Kp: Proportional gain (I*x*30)

Kd: Derivative gain (I*x*31)

Ki: Integral gain (I*x*33)

We will be looking at three key step response parameters to set the feedback:

Rise Time: the time it takes for the system to go from 10% to 90% of the commanded step (natural frequency is directly related to this)

Overshoot: the percentage past the commanded step that the system travels (damping ratio is directly related to this)

Settling time: the time it takes for the system to get and stay within 5% of the commanded step

What is typically desired is a quick rise time with little or no overshoot and quick settling time. The case of critical damping, which is the fastest possible rise time that creates no overshoot, is often the goal. There are usually tradeoffs between these parameters, particularly between fast response and low overshoot. If your amplifier has a tachometer, the tachometer is providing derivative gain (and therefore damping) within the amplifier itself. If the amplifier has been well tuned, you should not have to add any more derivative gain in the digital filter, but you are free to do so if you wish. On PMAC, it is possible to have the error integration active at all times by setting I*x*34 to 0, or to have it active only when the motion is stopped by setting I*x*34 to 1. While the step response for these two cases will look essentially identical, the behavior on real moves will be very different. Error integration that is active at all times can reduce following error on an extended profiled move, but at the cost of reduced system stability and of overshoot at the end of the move (which makes up for the lag at the beginning of the move). In a system without feedforward, the close following may be worth these costs. But the velocity and acceleration feedforward terms in PMAC can virtually eliminate following error without these drawbacks. For this reason, most PMAC customers use error integration only when motion is stopped -- where it can eliminate steady state errors due to static friction or net torque loads.

Because the proportional gain term Kp (I*x*30) is outside the brackets in the filter equation (see previous page), it also affects derivative and integral gains, and is not strictly speaking a true proportional gain. For this reason, if you modify Kp when Ki (I*x*33) or Kd (I*x*31) is not equal to zero, you are also changing the effective integral or differential gain. The shape of the response curve will not change much, although its timing will. You will want to change Ki and/or Kd in the opposite direction from Kp if you want to keep their effective gains constant.

Feedforward will affect step response even though it has no effect on the system stability we are really evaluating. Be sure that both acceleration and velocity feedforward are set to zero as you are doing the step responses.

The default step size of 100 encoder counts may or may be adequate. The guidelines are to make the step large enough so that the granularity of the position measurement is not a nuisance, but small enough so that the filter does not saturate on the step (the step size times proportional gain should be less than 178,950,000 with some margin; for instance a step size of 3000 with a proportional gain of 60000 will saturate, giving a misleading response).

Some systems will have mechanical resonance's in the coupling of the motor to the load. The PID filter cannot compensate for these resonance's; if their presence is not tolerable, you must keep the gains low enough not to stimulate them, or (preferably) stiffen the coupling to reduce the resonance's. (See the examples of a system with resonance, below, near the end.)

Doing the Step Response:

1. Set a safe starting filter with a little proportional gain, with no (or almost no) derivative or integral gain, and no feedforward. The current values for Kp, Ki, and Kd are displayed on the screen. See the figures on the following pages for sample values.

2. Do a step move and observe the plotted response displayed on the screen, along with the calculated statistics.

3. Adjust (probably increase) Kp (Proportional Gn) to get the fastest rise time possible without a huge amount of overshoot. Allow more overshoot here than you will in your final response.

4. Once you have a fast response, increase Kd (Derivative Gn) to bring down the overshoot to the desired value. Note that this will also increase the rise time.

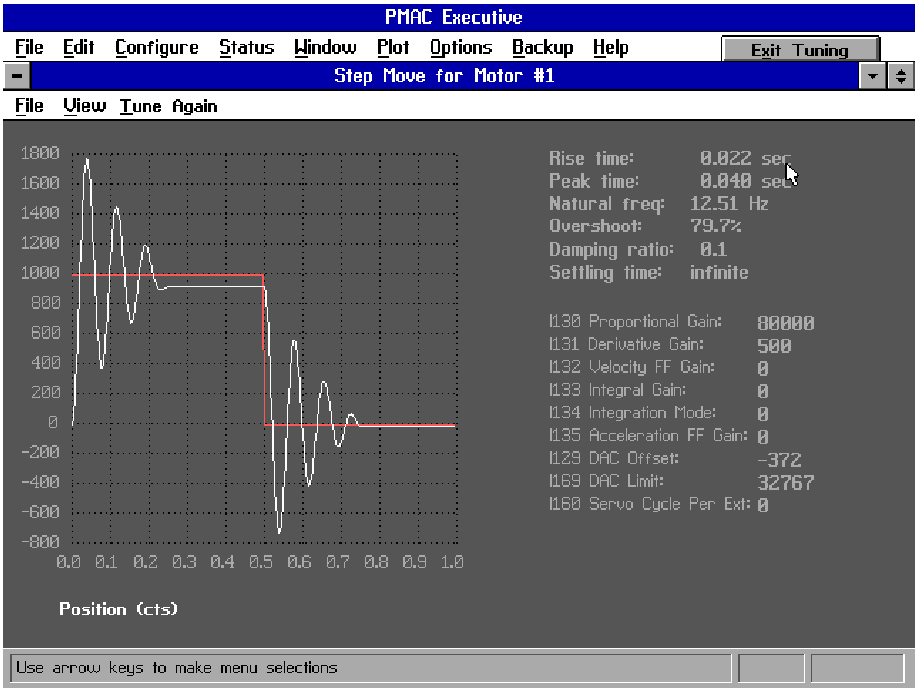
5. You may need to do further tradeoffs between Kp and Kd to get the desired response.

Note: You may wish to change the size and/or the duration of the step to be able to observe the response better. The default values are a 100 count step with a 500 millisecond dwell time.

6. Once you have set Kp and Kd, you have taken care of your dynamic step response (provided you are using error integration only in position). Now you will want to add integration to improve the static holding properties of the system. As you increase Ki (Integral Gn) and observe the step response, you will notice that it increases overshoot but comes back to the command position more quickly. A good value for Ki is one that brings the response back down to the command position as quickly as possible without going back past it.

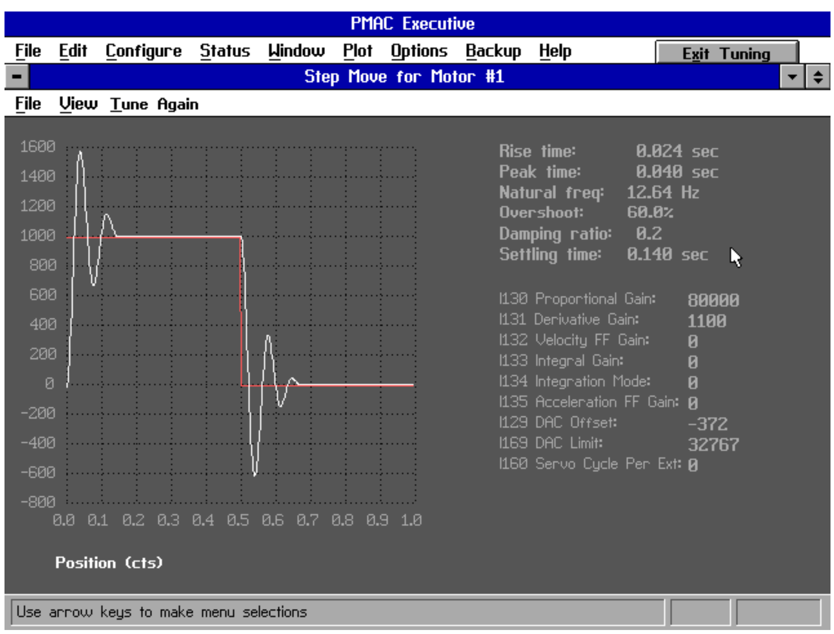
The following figures are taken from the actual tuning procedure for a motor using this program. They were done on a small DC laboratory motor with virtually no load, so the response is faster than it would be in almost any real-world application. The actual times are not important, however, the shapes of the response curves are. This system has a current-controlled amplifier with no tachometer. The goals of this system are critically damped step response, the quick elimination of steady-state errors at rest, and the minimization of following errors.

Initial step move response.



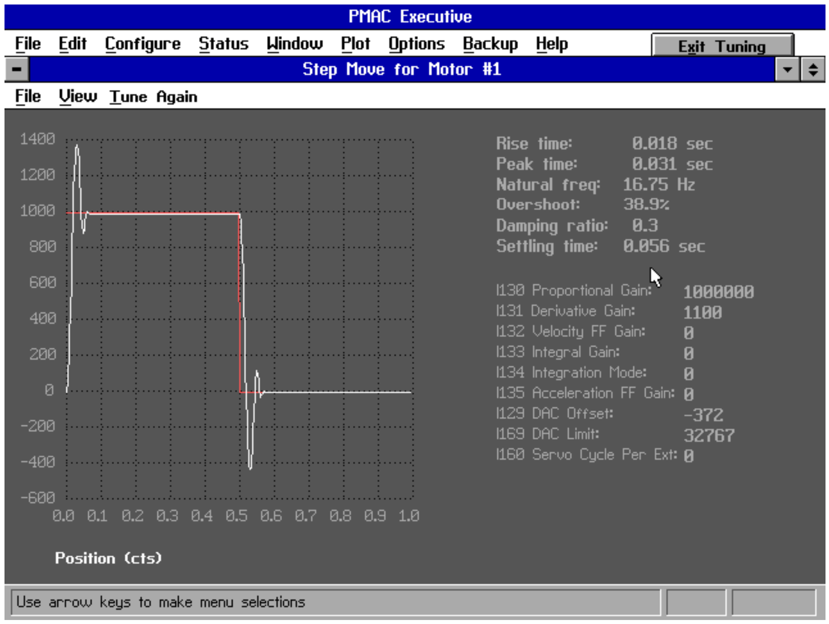
The screen on the previous page shows the step response with very little damping added (derivative gain Kd = 500). The large amount of ringing is obviously unacceptable; therefore some more Kd is needed. The next figure shows the response with an increased Kd of 1100. The ringing is largely eliminated; there is an overshoot of 60% and a rise time of 24 msec. Let us see if we can make the response quicker (which means a stiffer system).

Modified step response with higher Kd (Ix31).



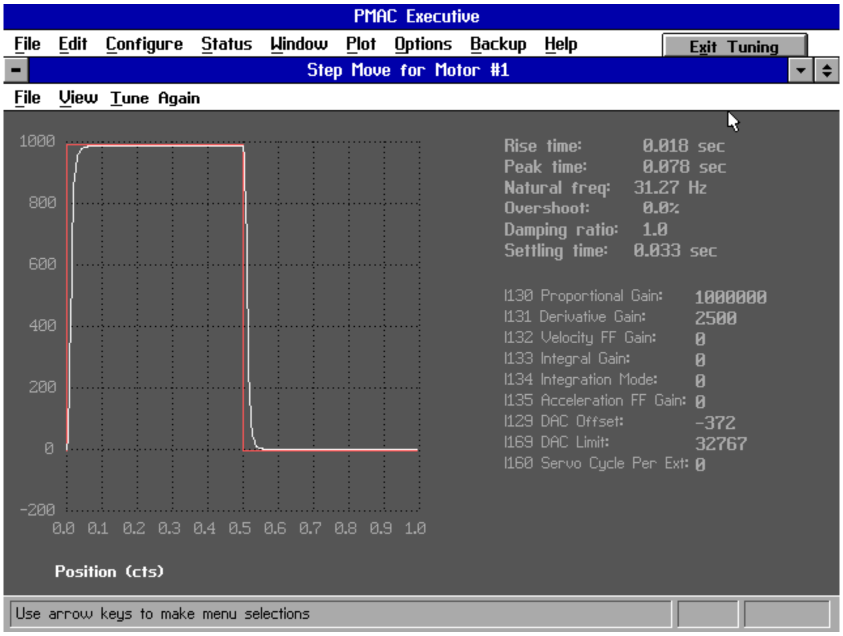
The next figure below shows the response with Kp increased from 80000 to 1000000. Note that the shape of the curve has not changed much (this is because the effective derivative gain is increasing with Kp), but the rise time has improved slightly (to 18 msec). We turn now to Kd.

Modified step response with higher Kp (Ix30).



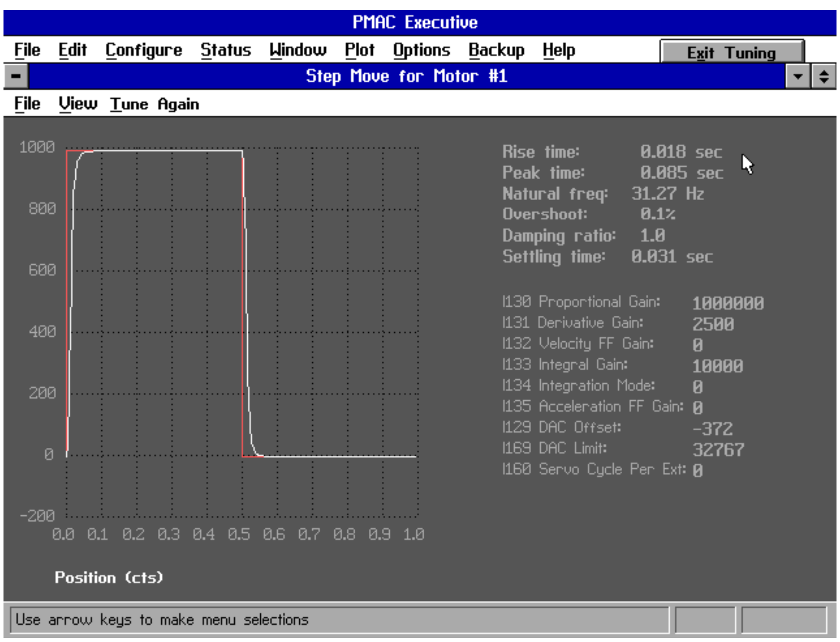
The figure below shows the step response with a Kd of 2500. This is the critically damped case; i.e. fast response with no overshoot. With Kd any smaller, we get some overshoot. And with it any larger, we just slow down the response. The tendency of the system to settle slightly off from the target position is due to a net torque or static friction. We will eliminate this with integral gain.

Modified step response with increased Kd again.



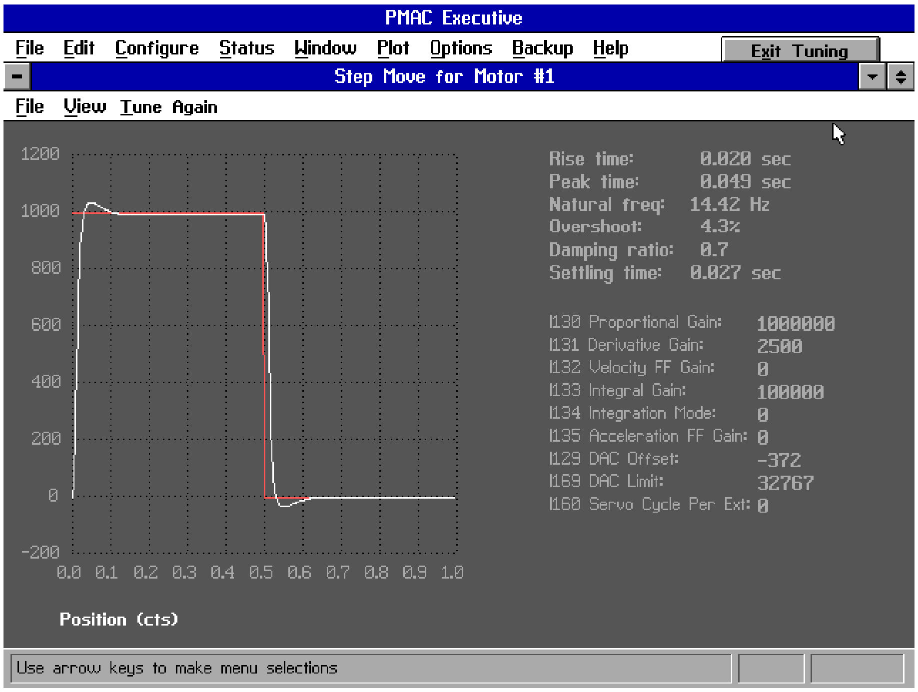
The figure below shows what happens with a little bit (relatively speaking) of integral gain (Ki = 10000): the steady- state error is gone, but the nature of the curve has not really changed.

Modified step response with increased Ki (Ix33).



The figure below shows the response curve for a substantially increased Ki (100000). This curve demonstrates how quickly the system would respond to a disturbance while in position. (Remember that we are using integral gain only when in position, so changing integral gain does not affect our actual dynamic response, although it will change the shape of the step response here. We still have the stability characteristics of critical damping while on the move.) Even the higher value of Ki does not result in oscillations, so we will use that. We have achieved what we wanted with feedback.

Modified step response with very high Ki.



Now we tackle feedforward...

Feedforward Tuning with a Profiled Parabolic Response

In a position servo system without feedforward or dynamic error integration, there must be a continual error between the commanded position and the actual position in a profiled move (known as following error) to produce a motor command. This means, however, that you are never really where you want to be when you want to be there, which is often the point of a profiled move in the first place. These following errors will usually be proportional (well correlated) to the velocity and/or the acceleration. The velocity and acceleration feedforward terms can be used to reduce these following errors virtually to zero. These parameters add terms to the torque command that are proportional to the commanded velocity and acceleration, respectively, in each cycle of the profiled move.

Mathematically speaking, if two sets of data, such as velocity and following error, vary in complete proportion to each other, they have a correlation of 1.0 (perfect correlation). If they vary completely independently of each other, they have a correlation of 0.0 (no correlation). The more they vary in proportion to each other, the closer their correlation will be to 1.0. In graphical terms, the more two curves are shaped like each other, the better they will be correlated. Another important figure is the constant of proportionality between the two sets of data, which is the average ratio between matching points in the sets. Even if two sets of data are very well correlated, the ratio may be so low that one of the sets is negligible in practical terms.

For each move done, the program will calculate the correlation between velocity and following error, and between acceleration and following error. It will also calculate the average ratio between following error and both velocity and acceleration. As a feedforward gain is increased, the ratio of following error to that quantity will decrease linearly (e.g. if a gain of 0 produces a ratio of 12.0, and a gain of 5000 produces a ratio of 6.0, then a gain of 10,000 can be expected to drive the ratio to zero). The ratio will decline even as the correlation stays high. When the ratio gets small enough, the correlation should decrease as some other factor becomes the dominant cause of following error (e.g. noise or the other feedforward gain). Ideally, the correlation will be brought near zero as well as the ratio.

PMAC has two feedforward terms:

Kvff: Velocity Feedforward gain (I*x*32)

Kaff: Acceleration Feedforward gain (I*x*35)

The strategy in this section is to do a series of "parabolic" moves (cubic in position, parabolic in velocity, linearly varying in acceleration), while adjusting the feedforward terms to reduce the following error and its correlation to velocity and acceleration. After each move, the program will automatically calculate the correlation's and ratios, and the maximum following error. These will be displayed on a plot of the position and the following error. The feedforward terms are increased from zero (working first with velocity feedforward) until the ratios (and hopefully the correlation's) are as close to zero as possible, without going strongly negative. If either correlation goes very far negative, you will be likely to get overshoot at the end of a move.

To get meaningful correlation information, particularly about acceleration, you must push the system hard. By increasing the length and/or decreasing the time of the move, you can get higher velocities and accelerations. Decreasing time is appropriate if increasing the length would cause problems with maximum travel or top velocity.

In a system without a tachometer, you will probably want to set the velocity feedforward equal to the derivative gain or slightly greater. In a system using a tachometer, the velocity feedforward should be greater than the derivative gain.

When you do the parabolic move without any feedforward, you will probably see a very high correlation to velocity (≈1.0) and almost no correlation to acceleration (≈0.0). There is most likely a real correlation to acceleration, but it is swamped out by the velocity correlation. As you reduce the velocity correlation, you should see increased acceleration correlation. When you then reduce the acceleration correlation, the correlation to velocity may increase again, but the actual ratio and magnitude of the following error should be very small.

Parabolic moves were chosen for this procedure because their acceleration and velocity vary continually and are uncorrelated to each other, making them ideal for this type of analysis. For further examination of the move, you may plot the velocity curve, the acceleration curve, or the following error curve. These are automatically scaled to fill up the plot window for maximum resolution. The move statistics are re-displayed with these plots.

Doing the Parabolic Move:

1. Do a parabolic move and observe the plotted response and the calculated statistics for the move.

2. Presuming there is a high correlation between following error and velocity (the velocity and following error curves have the same shape) and a relatively high maximum following error (as shown on the plot), increase Kvff (Velocity FF Gn I*x*31).

3. Do another parabolic move. If there is still inadequate Kvff, there will still be a high correlation, but the FE-to-Vel ratio and the maximum FE will be

reduced. Repeat steps 2 and 3 until you have the maximum Kvff that produces a square wave looking shape for the following error.

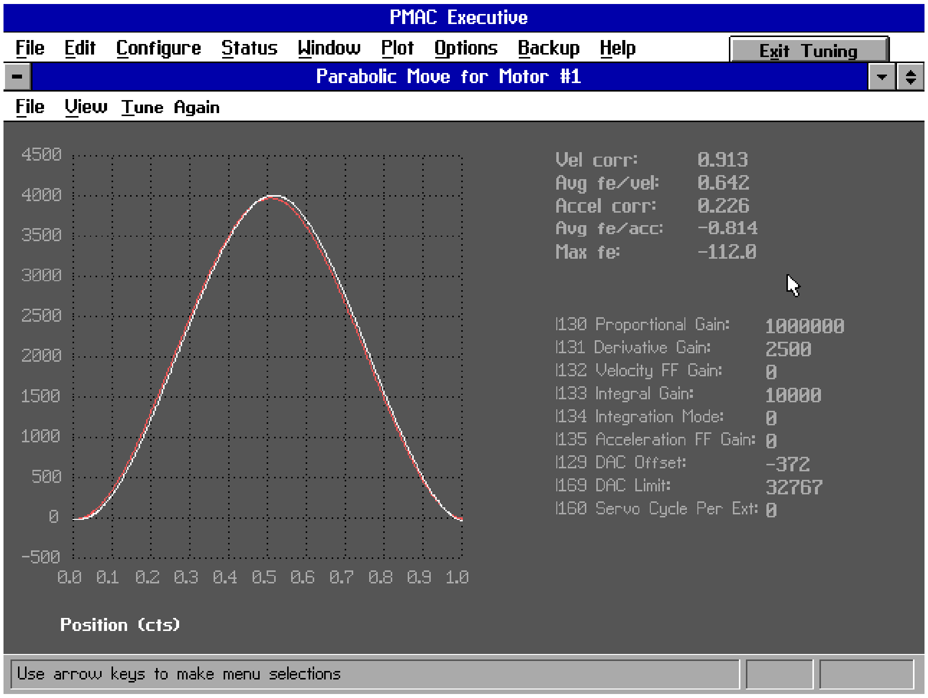
4. At this point, there should be noticeable correlation between acceleration and following error. You can see this by the numerical correlation value, or by plotting the acceleration and the following error, and noticing the similarity in shape. If you do not see any correlation, try increasing the length or decreasing the time of the move to get higher accelerations.

5. Increase Kaff (Accel FF Gn I*x*35). Do another move, and evaluate the correlation and FE-to-Accel ratio again. Repeat until you have the maximum Kaff that retains any positive correlation. At this point, you should have minimal following error, and most of what remains should be caused by noise or mechanical friction. If mechanical friction is the cause you will see a fairly strong velocity correlation because the friction causes the following error related to the sign of the velocity. However, each half of the following error curve should be quite flat if friction is the primary cause.

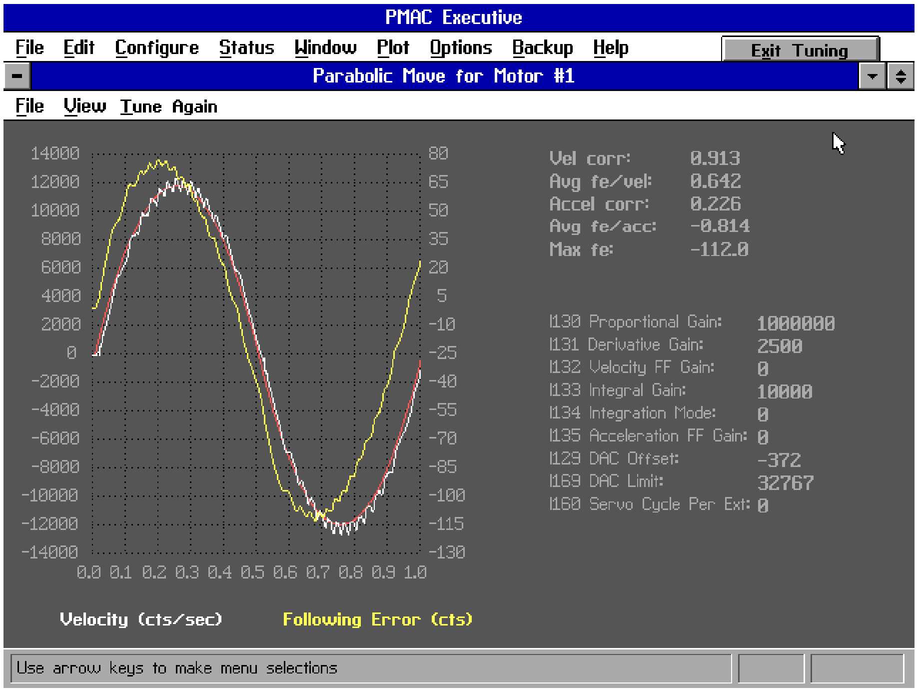
6. Pop open a beer. You are done!

In the first figure below, we see the plot of a parabolic move using a step size of 4000 counts and a step time of 500 milliseconds. Here, position is plotted against time, with the statistics of the following error shown. The following error gets as large as 112 counts, and it is virtually perfectly correlated to velocity (Vel corr = 0.913). It shows very low correlation to acceleration (Acc corr = 0.226). The next two figures show the plot of position, and velocity and following error, versus time for the same move. Note that the velocity and following error curves are shaped almost exactly alike. This is a ramification of their high correlation.

Initial parabolic move showing position.

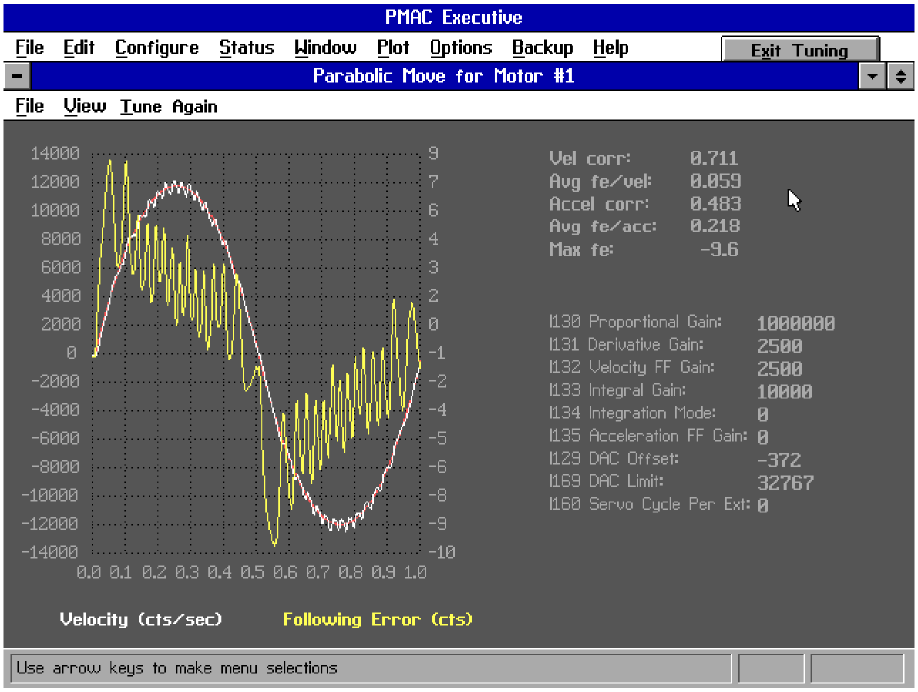

PMAC Executive
File Edit Configure Status Window Plot Options Backup Help
Exit Tuning
Parabolic Move for Motor #1
File View Tune Again
4500
4000
3500
3000
2500
2000
1500
1000
500
0
-500
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Position (cts)
Vel corr: 0.913
Avg fe/vel: 0.642
Accel corr: 0.226
Avg fe/acc: -0.814
Max fe: -112.0
I130 Proportional Gain: 1000000
I131 Derivative Gain: 2500
I132 Velocity FF Gain: 0
I133 Integral Gain: 10000
I134 Integration Mode: 0
I135 Acceleration FF Gain: 0
I129 DAC Offset: -372
I169 DAC Limit: 32767
I160 Servo Cycle Per Ext: 0
Use arrow keys to make menu selections

Initial parabolic move showing velocity and following error.

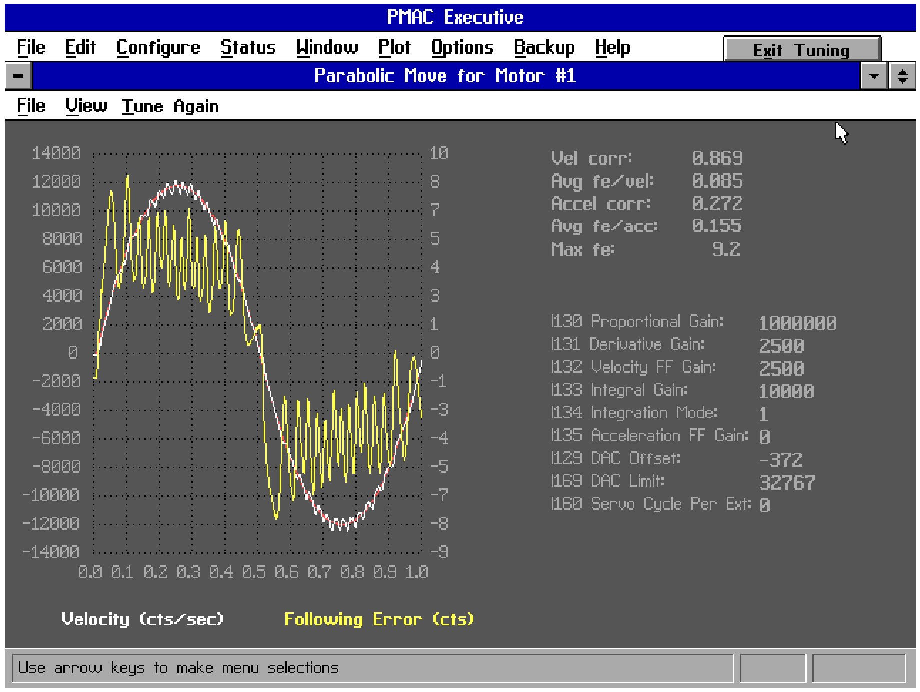


Now let's introduce some velocity feed forward. As a rule of thumb, choose Kvff (I*x*32) to be equal to Kd (I*x*31). In this case, Kvff = 2500. By looking at the following error curve, it no longer resembles the velocity curve (correlation of 0.211). Also, the maximum following error is reduced from 112 to 9.6 counts!

Modified parabolic move with increased Kvff (Ix32).

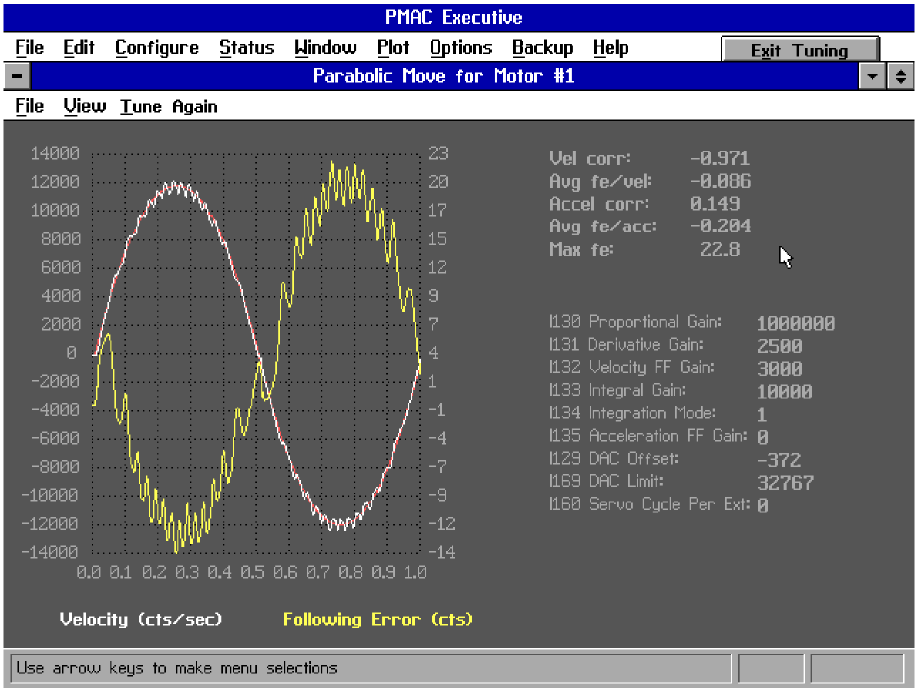

PMAC Executive
File Edit Configure Status Window Plot Options Backup Help
Exit Tuning
Parabolic Move for Motor #1
File View Tune Again
Vel corr: 0.711
Avg fe/vel: 0.059
Accel corr: 0.483
Avg fe/acc: 0.218
Max fe: -9.6
I130 Proportional Gain: 1000000
I131 Derivative Gain: 2500
I132 Velocity FF Gain: 2500
I133 Integral Gain: 10000
I134 Integration Mode: 0
I135 Acceleration FF Gain: 0
I129 DAC Offset: -372
I169 DAC Limit: 32767
I160 Servo Cycle Per Ext: 0
Velocity (cts/sec)
Following Error (cts)
Use arrow keys to make menu selections

Modified parabolic move with Ix34 = 1.

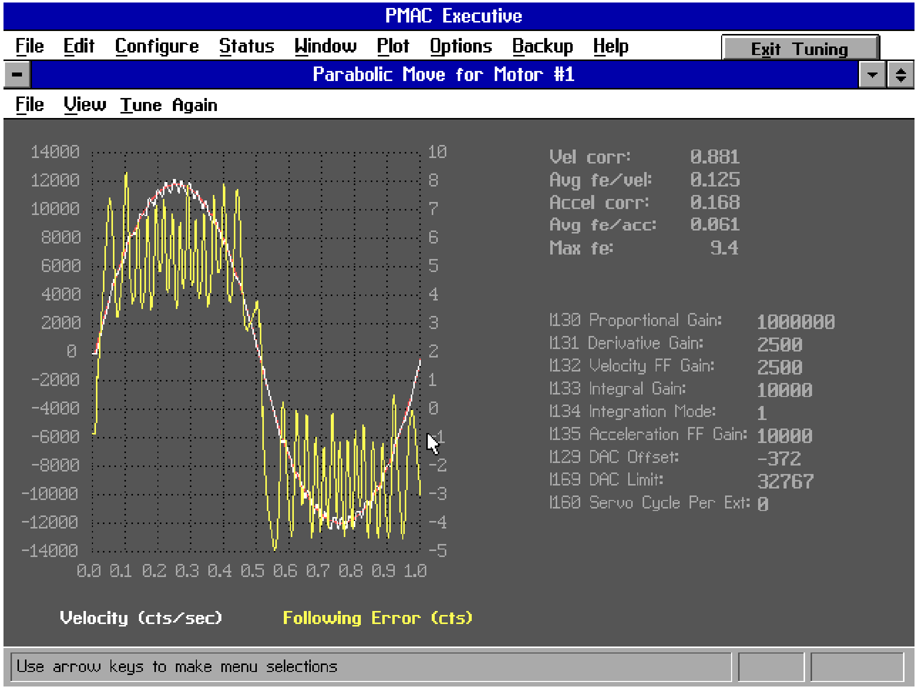


Now we instruct the integration in the servo loop to be active only when the motor is not be commanded to move. To do this, set Ix34, the integration mode, equal to 1. The improvement is seen above. If we continue to increase Kvff, then we have the following behavior where the following again resembles the velocity curve, but in an inverted manner.

Modified parabolic move with too much Kvff.


PMAC Executive
File Edit Configure Status Window Plot Options Backup Help
Exit Tuning
Parabolic Move for Motor #1
File View Tune Again
Vel corr: -0.971
Avg fe/vel: -0.006
Accel corr: 0.149
Avg fe/acc: -0.204
Max fe: 22.8
I130 Proportional Gain: 1000000
I131 Derivative Gain: 2500
I132 Velocity FF Gain: 3000
I133 Integral Gain: 10000
I134 Integration Mode: 1
I135 Acceleration FF Gain: 0
I129 DAC Offset: -372
I169 DAC Limit: 32767
I160 Servo Cycle Per Ext: 0
Velocity (cts/sec)
Following Error (cts)
Use arrow keys to make menu selections

Modified parabolic move with some Kaff.



Finally, a bit of acceleration feedforward, Kaff (I*x*35), gives us an almost perfectly tuned motor!